LING 408/508: Programming for Linguists

Lecture 27

December 9th
class ColorDieView(DieView):
    def setValue(self, value):
        self.value = value
        DieView.setValue(self, value)
    def setColor(self, self, color):
        self.foreground = color
        self.setValue(self.value)

Encapsulation  Separating the implementation details of an object from how the object is used. This allows for modular design of complex programs.

Polymorphism  Different classes may implement methods with the same signature. This makes programs more flexible, allowing a single line of code to call different methods in different situations.

Inheritance  A new class can be derived from an existing class. This supports sharing of methods among classes and code reuse.
Higher order functions

\textbf{map(function,list)}
apply function to each member of the list in turn
creating a transformed list

```
$ python
Python 2.7.10 (default, Aug 22 2015, 20:33:39)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.59.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> def square(x): return x**x
... >>> square(4)
16
>>> map(square,[1,2,3,4])
[1, 4, 9, 16]
```
Higher order functions

map(function, list)
apply function to each member of the list in turn
creating a transformed list

inline anonymous function: lambda
def example:
λx.x+x
lambda x:x+x

```python
>>> map(lambda x: x**x**x, [1, 2, 3, 4])
[1, 8, 27, 64]
```
Higher order functions

```python
reduce(function, list)
```

apply 2-argument function from left to right

1st argument will be cumulative value

2nd argument is fresh argument from the list

number of cannonballs in pyramid (Wikipedia):

```
>>> reduce(lambda x,y:x+y, map(square, [1,2,3,4,5]))
55
>>> map(square, [1,2,3,4,5])
[1, 4, 9, 16, 25]
>>> (((1+4)+9)+16)+25
55
```

\[
\frac{2n^3 + 3n^2 + n}{6}
\]
Higher order functions

```python
filter(function, list)
apply function to each member of the list
collect those for which the function returns true
```

```
>>> 10 % 4
2
[10 % 5]
0
[9 % 5]
4
[11 % 5]
1
[filter(lambda x:x%3, range(1,10))]
[1, 2, 4, 5, 7, 8]
[range(1,10)]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```
Standard Deviation Example

```python
import math
def mean(nums):
    return (reduce(lambda x,y:x+y,nums) + 0.0)/len(nums)

def mean(nums):
    return (reduce(sum,nums) + 0.0)/len(nums)

def stdDev(nums):
    m = mean(nums)
    sumsqs = reduce(lambda x,y:x+(y-m)*(y-m),nums)
    return math.sqrt(sumsqs/(len(nums)-1))

def sDev(nums):
    m = mean(nums)
    sum = 0.0
    for num in nums:
        dev = m - num
        sum += dev * dev
    return math.sqrt(sum/(len(nums)-1))

>>> l = [2,3,4]
>>> m = 3
>>> reduce(lambda x,y:x+(y-m)*(y-m),l)
3
>>> l = [0,2,3,4]
>>> reduce(lambda x,y:x+(y-m)*(y-m),l)
2
```
Median

- Median:
  - average of middle two if even number of elements
  - middle element of list if odd number of elements

```python
def median(nums):
    nums.sort()
    size = len(nums)
    midPos = size / 2
    if size % 2 == 0:
        median = (nums[midPos] + nums[midPos-1]) / 2.0
    else:
        median = nums[midPos]
    return median

>>> nums = [1,3,5,7]
>>> len(nums)/2
2
>>> nums[2]
5
>>> nums[2-1]
3
>>> nums = [1,3,5,7,9]
>>> len(nums)/2
2
>>> nums[2]
5```
Lists

• Comparisons:
  – default sort order: $\text{cmp}(x, y)$
  – returns $-1$ ($x < y$), $0$ ($x == y$), $1$ ($x > y$)
Lists

class Student:
    def __init__(self, name, hours, qpoints):
        self.name = name
        self.hours = float(hours)
        self.qpoints = float(qpoints)
    def getName(self):
        return self.name
    def getHours(self):
        return self.hours
    def getQPoints(self):
        return self.qpoints
    def gpa(self):
        return self.qpoints/self.hours

create a list of Student objects in variable data

def cmpGPA(s1, s2):
    return cmp(s1.gpa(), s2.gpa())

data.sort(cmpGPA)

An important point to notice here is that I did not put parentheses on the function name (cmpGPA()). I do not want to call the function. Rather, I am sending cmpGPA to the sort method, and it will call this function anytime it needs to compare two items to see what their relative ordering should be in the sorted list.

points, a “B” is 3 points, etc. Grade point averages are generally computed using quality points. If a class is worth 3 credit hours and the student gets an “A,” then he or she earns 3(4) = 12 quality points. To calculate a student’s grade point average (GPA), we divide the total quality points by the number of credit hours completed.
Dictionaries

• Hash table:
  ```python
  >>> passwd['guido']
  'superprogrammer'
  >>> passwd['bill']
  'monopoly'
  ```

• List:
  – delimiter: square brackets

• Dictionary:
  – delimiter: curly braces

```python
  >>> passwd = {'guido': 'superprogrammer', 'turing': 'genius', 'bill': 'monopoly'}
  >>> passwd['bill'] = 'bluescreen'
  >>> passwd
  {'turing': 'genius', 'bill': 'bluescreen', 'guido': 'superprogrammer'}
  ```
## Dictionaries

<table>
<thead>
<tr>
<th>Method</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;dict&gt;.has_key(&lt;key&gt;)</code></td>
<td>Returns true if dictionary contains the specified key, false if it doesn’t.</td>
</tr>
<tr>
<td><code>&lt;key&gt; in &lt;dict&gt;</code></td>
<td>Same as has_key</td>
</tr>
<tr>
<td><code>&lt;dict&gt;.keys()</code></td>
<td>Returns a list of the keys.</td>
</tr>
<tr>
<td><code>&lt;dict&gt;.values()</code></td>
<td>Returns a list of the values.</td>
</tr>
<tr>
<td><code>&lt;dict&gt;.items()</code></td>
<td>Returns a list of tuples (key, value) representing the key-value pairs.</td>
</tr>
<tr>
<td><code>&lt;dict&gt;.get(&lt;key&gt;, &lt;default&gt;)</code></td>
<td>If dictionary has key returns its value; otherwise returns default.</td>
</tr>
<tr>
<td><code>del &lt;dict&gt;[&lt;key&gt;]</code></td>
<td>Deletes the specified entry.</td>
</tr>
<tr>
<td><code>&lt;dict&gt;.clear()</code></td>
<td>Deletes all entries.</td>
</tr>
</tbody>
</table>

```python
>>> passwd.keys()
['turing', 'bill', 'newuser', 'guido']
>>> passwd.values()
['genius', 'bluescreen', 'ImANewbie', 'superprogrammer']
```
Dictionaries

• Word frequency table program: tokenization

```python
fname = raw_input("File to analyze: ")

# read file as one long string
text = open(fname, ’r’).read()

# convert all letters to lower case
text = string.lower(text)

# replace each punctuation character with a space
for ch in ’!"#$%&()*.+-/:;<=?@[\]^_`{\|}~’:
    text = string.replace(text, ch, ’ ’)

# split string at whitespace to form a list of words
words = string.split(text)
```
Dictionaries

- Word frequency table program

```python
counts = {}
for w in words:
    counts[w] = counts.get(w, 0) + 1

# get list of words that appear in document
uniqueWords = counts.keys()

# put list of words in alphabetical order
uniqueWords.sort()

# print words and associated counts
for w in uniqueWords:
    print w, counts[w]
```

default = 0
Dictionaries

• Word frequency table program

items = counts.items()

Here items will be a list of tuples (e.g., [('foo',5), ('bar',7), ('spam',376), ...]).

def compareItems((w1,c1), (w2,c2)):
    if c1 > c2:
        return -1  # items.sort(compareItems)
    elif c1 == c2:
        return cmp(w1, w2)
    else:
        return 1
Corpus

• Lead NYTimes article one year ago tomorrow:

WASHINGTON — A scathing report released by the Senate Intelligence Committee on Tuesday found that the Central Intelligence Agency routinely misled the White House and Congress about the information it obtained from the detention and interrogation of terrorism suspects, and that its methods were more brutal than the C.I.A. acknowledged either to Bush administration officials or to the public.

The long-delayed report, which took five years to produce and is based on more than six million internal agency documents, is a sweeping indictment of the C.I.A.'s operation and oversight of a program carried out by agency officials and contractors in secret prisons around the world in the years after the Sept. 11, 2001, terrorist attacks. It also provides a macabre accounting of some of the grisliest techniques that the C.I.A. used to torture and imprison terrorism suspects.

Detainees were deprived of sleep for as long as a week, and were sometimes told that they would be killed while in American custody. With the approval of the C.I.A.'s medical staff, some C.I.A. prisoners were subjected to medically unnecessary “rectal feeding” or “rectal hydration” — a technique that the C.I.A.'s chief of interrogations described as a way to exert “total control over the detainee.” C.I.A. medical staff members described the waterboarding of Khalid Shaikh Mohammed, the chief planner of the Sept. 11 attacks, as a “series of near drownings.”
The solution is to use Python's raw string notation for regular expression patterns; backslashes are not handled in any special way in a string literal prefixed with `r`. So `r"\n"` is a two-character string containing `\` and `\n`, while `"\n"` is a one-character string containing a newline. Usually patterns will be expressed in Python code using this raw string notation.
Results

dhcp-10-142-129-92:ling508-15 sandiway$ python wc.py article.txt 25
the 163
, 102
. 81
of 70
to 67
to 67
C.I.A. 57
that 56
and 54
in 43
a 39
" 31
" 30
report 27
The 26
for 20
by 19
were 19
said 17
was 16
- 16
as 14
had 13
at 12
it 12
program 12

sys.argv[1]

sys.argv[2]

top 2 content words in the article
Adminstrivia

- TCEs are fully online (paper TCEs are history)

<table>
<thead>
<tr>
<th>Class</th>
<th>Tce Start/End Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC 538 - 001</td>
<td>11/16/2015 to 12/9/2015 11:59:00 PM</td>
</tr>
<tr>
<td>CSC 438 - 001</td>
<td>11/16/2015 to 12/9/2015 11:59:00 PM</td>
</tr>
<tr>
<td>LING 508 - 001</td>
<td>11/16/2015 to 12/9/2015 11:59:00 PM</td>
</tr>
<tr>
<td>LING 438 - 001</td>
<td>11/16/2015 to 12/9/2015 11:59:00 PM</td>
</tr>
<tr>
<td>LING 538 - 001</td>
<td>11/16/2015 to 12/9/2015 11:59:00 PM</td>
</tr>
<tr>
<td>LING 408 - 001</td>
<td>11/16/2015 to 12/9/2015 11:59:00 PM</td>
</tr>
</tbody>
</table>

https://tce.oirps.arizona.edu/TCEOnline